Workung with large gridded data sets - formats and access methods

Martin Schmidt
Leibniz-Institute for Baltic Sea Research Warnemuende
martin.schmidt@io-warnemuende.de

Distributing or accessing huge data sets

- satellite data
- model results

Problem: data volume TByte – Pbyte

“I use to keep all data on my laptop ...”

How to publish such huge data sets?

Ask for help:
"Could you please kindly provide that image with another label?"

Work together!
Share the data!

Do it Yourself!

Motivation

Complete switch from paper based data storage to digital data storage

Growing amount and complexity of data to be managed for scientific work (satellite data, input and results of numerical models)

International and interdisciplinary data exchange

Different data formats between programming languages, architectures

- need for separate read/write procedures for each data set
- incomplete metadata
- error prone

Unidata's network Common Data Form

UCAR COMMUNITY PROGRAMS | unidata Data Services and Tools for Geoscience

Requirements

- suitable for data defined on a grid
- *selfdescribing*: includes data and metadata defining the meaning of the data, keep data and metadata strictly together
- *scalable*: easy and fast access to subsets
- *portable*: platform and system independent treatment of data, follow standards and conventions
- *sharable*: read and write of many processes possible the same time (supercomputing)
- *compatible*: early data versions stay supported
- *efficient* storage (compression, packing)

Unidata's network Common Data Form

Craetors

Glenn Davis, Russ Rew, Ed Hartnett, John Caron, Dennis Heimbigner, Steve Emmerson, Harvey Davies, and Ward Fisher at the Unidata Program Center in Boulder, Colorado

Origin

NASA's CDF data model

Home

University Corporation for Atmospheric Research.

Funding

NSF

Unidata's network Common Data Form

UCAR COMMUNITY PROGRAMS | unidata Data Services and Tools for Geoscience

Realisation

- Read/write access through a well defined interface
  netCDF-library.
- Easy install, documented, support
- available for all systems (UNIX, linux, mac, windows, android)
  (from supercomputer to smartphone)
- interface for many programming and scripting languages
  FORTRAN, C, C++, python, java, octave (matlab)
- implemented in most data processing and visualisation tools

Climate and Forecast metadata conventions

Self-describing language is impossible. Conventions on the meaning of elements are needed.

Data are valid in the context of a grid.

Coordinates are strictly monotonic!

- latitude, longitude
- depth, pressure density
- time
- forecast time
- ensemble number

.....

Climate and Forecast metadata conventions

Metadata

- part of the data file structure, no external files.
- units should use SI-conventions
- readable by humans and machines (programs)

Climate and Forecast metadata conventions

Dimensions and coordinates

- define an index space
- coordinates are independent variables
  (lat, lon, distance depth, pressure, time, forecast time, ensemble)
- appropriate time usage (calendar, beware of years and months)

Data:

- defined on grid spanned by the dimensions
- depend on coordinates
- cell bounds and cell method can be addressed

```
ncdump -h hix_data.nc
netcdf hix_data {
dimensions:
        altitude = 1 ;
        latitude = 1 ;
        longitude = 1 ;
        time = 1529 ;
variables:
        float longitude(longitude) ;
                longitude:units = "degree_east" ;
                longitude:point_spacing = "even" ;
                longitude:axis = "X" ;
                longitude:standard_name = "Longitude" ;
        float latitude(latitude) ;
                latitude:units = "degree_north" ;
                ….
        double time(time) ;
                time:units = "days since 1800-01-01 00:00:00" ;
                time:axis = "T" ;
                time:calendar = "GREGORIAN" ;
                time:time_origin = "01-Jan-1800" ;
                time:standard_name = "Time" ;
        float HIX(time, latitude, longitude) ;
                HIX:units = " " ;
                HIX:long_name = "St. Helena Island Climate Index (HIX)" ;
        float SLP(time, latitude, longitude) ;
                SLP:units = "hPa" ;
                SLP:long_name = "Sea level pressure" ;
```

File structure internal compression, chunking

Direct and strided access

- single or groups of data items can be addressed directly

Internal compression

- no packed and unpacked file versions any more
- chunking

Access through a MPI-layer

- parallel writing from many processes → supercomputer application

Remote access

How to share Tbyte to Pbyte of data?

Local file server?

- duplicate and double data sets, build local data bases
- hardware and energetic costs, (wo)manpower
- legal issues, data ownership?

Remote distributed storage

- keep data sets local, avoid duplication and transfer
- distributed data bases
- keep responsibility and control over data

Unidata's Thematic Real-time Environmental Distributed Service

UCAR COMMUNITY PROGRAMS | unidata Data Services and Tools for Geoscience

Concept

- keep data on a server → THREDDS data server, TDS
- web access over standard, widespread web services
- searchable
- keep data and metadata strictly together
- use a markup language (ncml) to serve all data in a unified standard data format (netCDF – free supported for all platforms)
- a data set can be a reference to another TDS → build distributed data bases, share data without copying and multiplying PByte

OpenDAP Open Source project for a Network Data Access Protocol

Concept

- libdap: provides web access to netCDF formated data over a web interface
- user view: use an URL as filename
- part of the netcdf library
- all software using the netCDF library inherits this capability

Example data set access

Access for catalog browsing

- web-browser https://thredds-iow.io-warnemuende.de
- click through the catalog entries
- find the data set you are interested in
- inspect the services available for this data set
- find the URL for data access (retrieval)

Example data set access

TdsStaticCatalog http://thredds.io-warnemuende.de:8080/thredds/catalog.html - Mozilla Firefox

File Edit View History Bookmarks Tools Help

CurrentMachineload < ... | IceWM Guide | Celettu'... | TdsStaticCatalog http://... | Unidata | THREDDS Dat... | ThesisSeminar - Files - ... | Bildschirmfotos › Wiki › ...

thredds.io-warnemuende.de:8080/thredds/catalog.html

linux screeshots

Most Visited | Novell | Getting Started | Latest Headlines | Mozilla Firefox | Leo fragen | Delta-Distribution – Wi...

Catalog http://thredds.io-warnemuende.de:8080/thredds/catalog.html

Dataset	Size	Last Modified
Global Data		--
===> Global Data THREDDS Catalog/		--
Baltic Sea Data		--
===> Baltic Sea Data THREDDS Catalog/		--
===> MARNET Data THREDDS Catalog/		--
South Atlantic Data		--
===> South Atlantic Data THREDDS Catalog/		--
Persian Gulf Data		--
===> Persian Gulf Data THREDDS Catalog/		--
Data sets for modelling with MOM-3		--
Data sets for modelling with MOM-4		--
Data sets for modelling with GETM		--
===> GETM modelling THREDDS Catalog/		--
Model results		--
===> Baltic Sea MOM Models THREDDS Catalog/		--
===> Baltic Sea GETM Models THREDDS Catalog/		--
===> South Atlantic Models THREDDS Catalog, GENUS ecosystem experiments/		--
===> South Atlantic Models THREDDS Catalog, PREFACE physics experiments/		--
Other data sets		--
===> Benguela related CTD-sections/		--
===> Benguela moorings/		--
===> Ship borne underway data/		--
Test Enhanced Catalog/		--

Initial TDS Installation at My Group see Info
THREDDS Data Server [Version 4.3.23 - 20140826.1617] Documentation

About NCAR & UCAR | UCAR | Catalog Services

thredds.io-warnemuende.de:8080/thredds/genus/catalog_exp19_run2.html?dataset=IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16

update windows 7 windows10

Initial TDS Installation

THREDDS Data Server

Catalog http://thredds.io-warnemuende.de:8080/thredds/genus/catalog_exp19_run2.html

Dataset: IOW THREDDS Server Catalog/South Atlantic Ecosystem model, CCMP-2 winds EXP 19, RUN 2, ocean data, 5-day time averages

- *Data type:* GRID
- *Naming Authority:* IOW
- *ID:* IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16

Documentation:

- **summary:** "Tropical and subtropical Atlantic ecosystem model. CCMP-2 forcing, Chlorophyll from ERGOM, solar radiation reduced - using Bodin formula"
- Description Web page
- IOW documentation

Access:

1. **OPENDAP:** /thredds/dodsC/genus/IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16.nc
2. **ISO:** /thredds/iso/genus/IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16.nc
3. **NCML:** /thredds/ncml/genus/IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16.nc
4. **UDDC:** /thredds/uddc/genus/IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16.nc
5. **WMS:** /thredds/wms/genus/IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16.nc
6. **HTTPServer:** /thredds/fileServer/genus/IOW-THREDDS-genus_exp19_run2_5day_ocean_2017-10-20-16.nc

Keywords:

- modelling, subtropical atlantic, tropical atlantic, upwelling, Benguela

Projects:

- SACUS **(DIF)**

Properties:

- Institute = "Leibniz Institute for Baltic Sea Research"
- contributor = "Martin Schmidt"
- creator = "Martin Schmidt"

Viewers:

- Godiva2 (browser-based)
- NetCDF-Java ToolsUI (webstart)
- Integrated Data Viewer (IDV) (webstart)

Zur Suche Text hier eingeben

About NCAR & UCAR | UCAR | OPeNDAP Dataset Query Fo

thredds.io-warnemuende.de:8080/thredds/dodsC/genus/genus_run_88_5day_ocean.nc.html

67% update windows 7 windows10

OPeNDAP Dataset Access Form

Tested on Netscape 4.61 and Internet Explorer 5.00.

Action: Get ASCII | Get Binary | Show Help

Data URL: http://thredds.io-warnemuende.de:8080/thredds/dodsC/genus/genus_run_8

Global Attributes:

```
filename: RUN.2012.04.12.00/ocean_day.nc
title: MOM4 Baltic Sea 3 n.m.
grid_type: regular
grid_tile: N/A
```

Variables:

xu_ocean: Array of 64 bit Reals [xu_ocean = 0..272]

xu_ocean:

```
long_name: ucell longitude
units: degrees_E
cartesian_axis: X
_ChunkSize: 273
```

yu_ocean: Array of 64 bit Reals [yu_ocean = 0..381]

yu_ocean:

```
long_name: ucell latitude
units: degrees_N
cartesian_axis: Y
_ChunkSize: 382
```

nv: Array of 64 bit Reals [nv = 0..1]

nv:

```
long_name: vertex number
units: none
cartesian_axis: N
_ChunkSize: 2
```

xt_ocean: Array of 64 bit Reals [xt_ocean = 0..272]

xt_ocean:

```
long_name: tcell longitude
units: degrees_E
cartesian_axis: X
_ChunkSize: 273
```

yt_ocean: Array of 64 bit Reals [yt_ocean = 0..381]

yt_ocean:

```
long_name: tcell latitude
units: degrees_N
cartesian_axis: Y
_ChunkSize: 382
```

st_ocean: Array of 64 bit Reals [st_ocean = 0..88]

The URL

Zur Suche Text hier eingeben

Example data set access

Access for data retrieval

- web browser to retrieve data? Works but not recommended
- open the data set with your beloved visualisation program, use URL as file name

- Behind the scene:

  your tool must be able to read netCDF-files openDAP access must be enabled

  (Open-source Project for a Network Data Access Protocol)

  Excel is not able to read netcdf, strong reason to learn something else

Example data set access

Software examples

Free software for reading netcdf files

https://www.unidata.ucar.edu/software/netcdf/software.html

- **Linux**: ferret, R, grADS, GMT, cdo, nco, python, ncview, octave ….
- **Windows:** R, python, octave

Commercial packages (may be expensive!)

- Matlab, IDL, Surfer

Example for TDS data access: POC from MODIS

Access with ferret:

step 1: find the data set URL in the THREDDS catalog (web browser)

http://oceanwatch.pfeg.noaa.gov/thredds/catalog.html

→ *http://oceanwatch.pfeg.noaa.gov/thredds/dodsC/satellite/MPOC/mday*

step 2: start *ferret* and open the data set and have a look at the data set structure

Yes? use https://thredds-iow.io-warnemuende.de/thredds/dodsC/balticA1B_3nm/joined_ocean.nc

yes? show data
yes?
1>

yes? shade/x=10:16/y=-28:-10 /l=100 POC

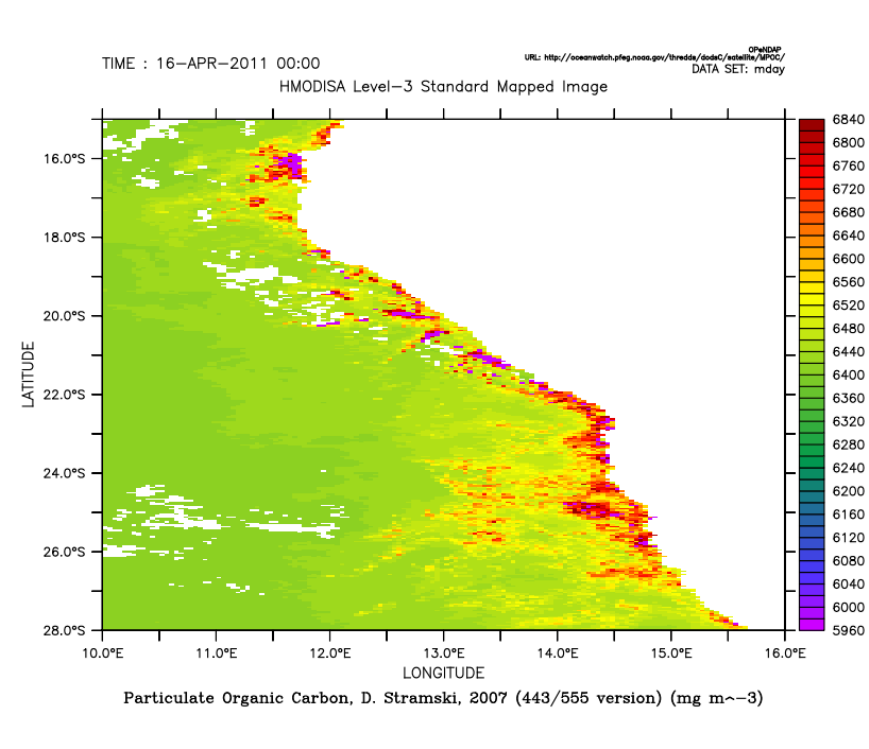
netcdf-bug: use [fillmismatch]http://oceanwatch.pfeg.noaa.gov/thredds/dodsC/satellite/MPOC/mday

Example for TDS data access

step 3: make an example plot (for the Benguela upwelling system)

yes? shade/x=10:16/y=-28:-10 /t=16-apr-2011 POC
yes? frame/file=poc_example.png

step 4: scientific work with the data



Example for TDS data access: Model results

Access with R:

step 1: find the data set URL in the THREDDS catalog (web browser)

https://thredds-iow.io-warnemuende.de/thredds/catalogs/regions/baltic/climate_projections/catalog_climate_projections.html

step 2: start *R* and open the data set and show content

```
>library(ncdf4)
>file.1 <- "https://thredds-iow.io-warnemuende.de/thredds/dodsC/balticA1B_3nm/joined_ocean.nc"
>nc.1 <- nc_open(file.1)
>print(nc.1)
```

Example for TDS data access: Model results

Remarks – sophisticated subsetting:

Data set size about 5GByte

Only the netCDF header was read.

For the plot 200 Kbyte are transferred.

Summary

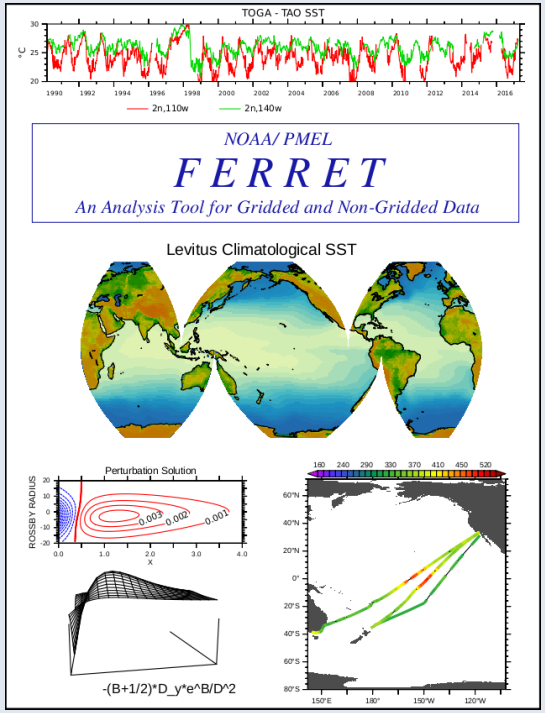
- **huge data sets like model results or satellite data can be distrubuted via a THREDDS data server (TByte – PByte)**
- **a data set in THREDDS can be a reference to another data set in another THREDDS build linked data bases**
- **metadata from model results on THREDDS can be added to metadata base**

Science Data Integration Group

Ferret Support

Pacific Marine Environmental Laboratory

Data visualisation and analysis with Ferret



authors: Thermal Modeling and Analysis Project (TMAP)

PMEL in Seattle

S. Hankin, D.E. Harrison, J. Sirott, A. Manke,

J. Callahan, J. Davison, K. O'Brien ….

Operating systems: X11 based (Linux, MacOS)

(Windows → virtual machine)

Download, support, license, installation:

https://ferret.pmel.noaa.gov/Ferret/

Open Source Definition

statically linked binaries

build it yourself (prerequisite netcdf)

ongoing python integration

Getting startet

environment: before ferret starts set some environment variables

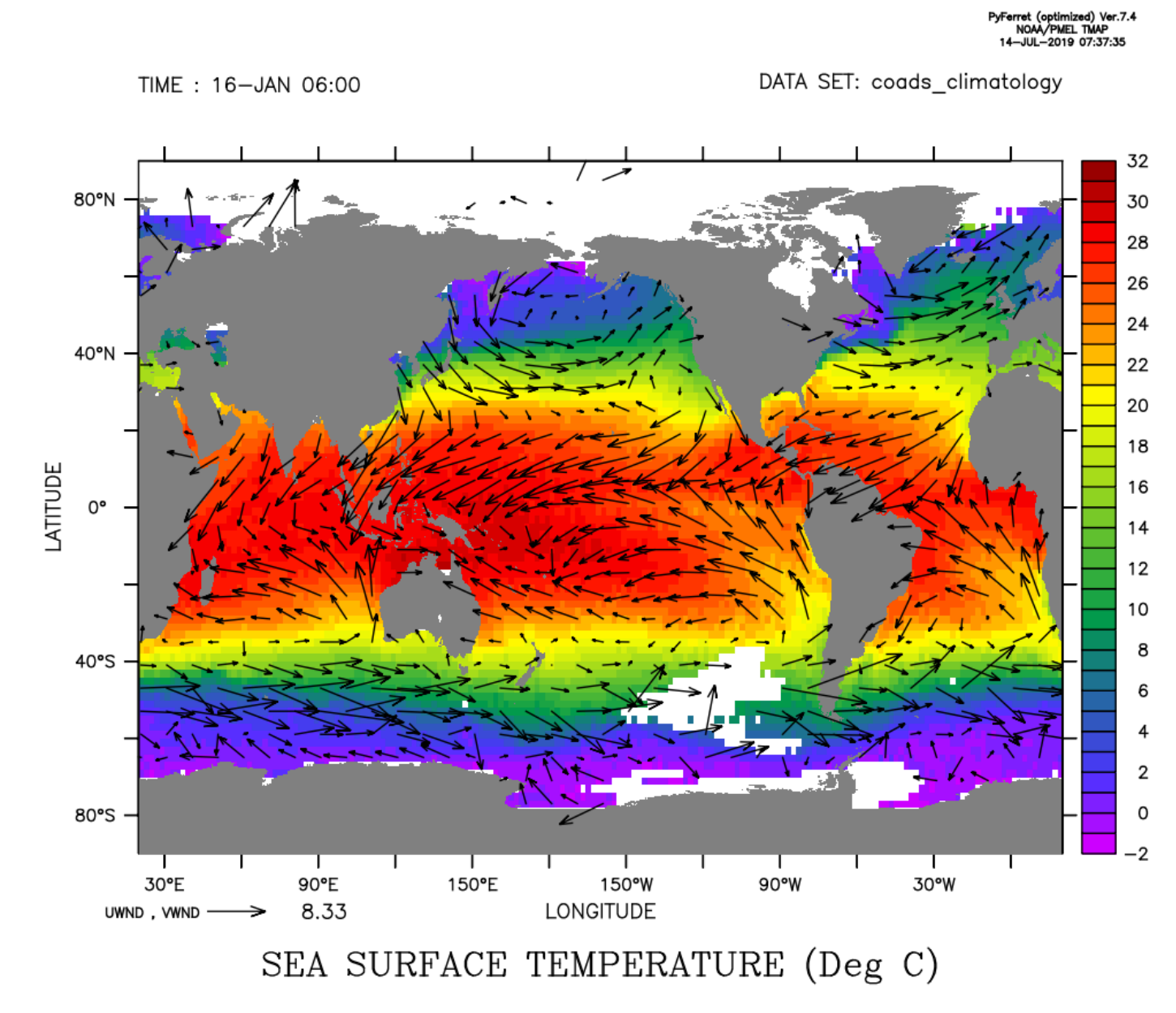
FER_DIR, FER_PATH, FER_DSETS ….

command line start:

```
> ferret
NOAA/PMEL TMAP
        PyFerret v7.4 (optimized)
        Linux 4.4.126-48-default - 04/29/18
        14-Jul-19 07:32
yes?
```

Check your environment:

```
yes? use coads_climatology
yes? show data
yes? shade/l=1 sst
yes? go fland 20
yes? vec/over/l=1 uwnd,vwnd
yes? frame/file=sst_wind_jan.png/xpixels=1024
```


PyFerret (optimized) Ver.7.4
NOAA/PMEL TMAP
14-JUL-2019 07:37:35
TIME : 16-JAN 06:00
DATA SET: coads_climatology
80°N
40°N
0°
40°S
80°S
LATITUDE
30°E
90°E
150°E
150°W
90°W
30°W
LONGITUDE
UWND , VWND
8.33
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
−2
SEA SURFACE TEMPERATURE (Deg C)

What is special?

data set analysis when opening:

explores the data set automatically uning netcdf features

units:

output has units, taken from the netcdf input file

coordinates and data together:

output is in geographical coordinates

dense command syntax:

one line for a figure

→ reasonable defaults

Context dependent variables:

```
yes? shade/l=1 sst
yes? shade/x=0 sst
yes? plot/x=0/y=0 sst
yes? list/l=12/x=10/y=-23:-15 sst
```

Accessing data?

files:

yes? use <filename>

DAP-datasets:

yes? use https://thredds-iow.io-warnemuende.de/thredds/dodsC/genus/IOW-THREDDS-genus_exp19_run3_5day_ocean_2017-10-20-16.nc

file sets with aggregation:

yes? go aggregate sst_*.nc 1

Show the available data sets:

yes? Show data

yes? Show /brief data

yes? stat temp

Using your own coordinates?

See coordinates of variables:

```
yes? list x[gx=temp]          ! show zonal coordinate
yes? list t[gt=temp]          ! show time coordinate
```

To serve as a coordinate, a sequence must be monotonic!

Define axes:

```
yes? define axis/x=-10:16:1/unit=degrees_east lons
yes? define axis/y=-28:6:1/unit=degrees_north lats
yes? define axis/t=1-jan-2010:1-jan-2011:1/unit=days/t0=1-jan-2000/cal=gregorian times
```

Define a new grid:

Keep the original time and vertical axis!

```
yes? define grid/like=temp/x=lons/y=lats new_grid
```

Put velocity vector onto the coarser grid for plotting:

```
yes? let u_1 = u[g=new_grid@ave]; let v_1 = v[g=new_grid@ave];
yes? shade/t=1-jan-2011/k=1 (u^2+v^2)^.5
yes? vec/over/k=1/t=1-jan-2011 u_1, v_1
```

Make a climatology

See climatological axes:

```
yes? sh axis/all
name                    axis     # pts  start          end
MONTH_IRREG             TIME     12mi  16-JAN 12:00   15-DEC 17:49
MONTH_REG               TIME     12mr  16-JAN 06:00   16-DEC 01:20
SEASONAL_REG            TIME     4mr   15-FEB 15:43   15-NOV 14:05
MONTH_GREGORIAN TIME    12mi  16-JAN 12:00   15-DEC 17:49
MONTH_NOLEAP            TIME     12mi  16-JAN 12:00   16-DEC 12:00
MONTH_360_DAY           TIME     12mr  16-JAN 00:00   16-DEC 00:00
MONTH_ALL_LEAP          TIME     12mi  16-JAN 12:00   16-DEC 12:00
MONTH_JULIAN            TIME     12mi  16-JAN 12:00   15-DEC 18:00
```

To serve as a coordinate, a sequence must be monotonic!

Use the modulo-transformation:

```
yes? say `sst,return=calendar`
yes? let sst_climatology = sst[gt=month_julian@MOD]
```

Make a climatology

Use the modulo-transformation:

yes? say `sst,return=calendar`

yes? let sst_climatology = sst[gt=month_julian@MOD]

Find the anomaly:

yes? let sst_anomaly = sst - sst_climatology[gt=sst@lin]

Make a plot for the Benguela system (off Walvis Bay):

yes? plot/x=13.5/y=-23 sst_anomaly

Be careful with the discussion of the result! Be aware, the model is a forced one and forced with CCMP2.

Analyse data on density surfaces

Use the @weq-transformation:

```
yes? let  sig262 = sig[z=@WEQ:26.2]
yes? let sig = rho_un(salt,temp,0) - 1000
yes? shade/l=1/x=0/z=0:800 sig262     ! see the mask properties
yes? shade/l=1 sig262[k=@sum]         ! see the completeness
```

Find temperature and salinity on these cells:

```
yes? let temp262 = temp*sig262
yes? shade/l=1 temp262[k=@sum]
```

Make a plot for the Benguela system (off Walvis Bay):

```
yes? shade/l=1/x=5:16/y=-25:-15 temp262[k=@sum]
```